Use of External Standards within NIEM

Georgia Tech Research Institute

March 28, 2006

Introduction

The primary goal of this paper is to provide consistent guidelines for NIEM users wishing to profile and use external standards with NIEM. In many cases employing particular standards in a NIEM Information Exchange Package Description (IEPD) may actually be preferred.

There are a variety of commonly used standards that are currently represented in XML Schema. There must be a method for NIEM to promote and use these external standards where requirements dictate.

There are two fundamental cases.

- Case 1: NIEM IEPDs reference, import, and use components in an external standard schema or namespace that does not conform to NIEM Naming and Design Rules.
- Case 2: External schemas reference, import, and use NIEM components.

This paper focuses on case 1. It presents a methodology for including non-NIEM components in NIEM-conformant schemas. It enables data modeling efforts to build NIEM-conformant components from non-NIEM data objects.

Case 2 is outside the scope of this paper. However, it should be noted that the use of NIEM in external standards will likely involve per-standard recommendations. Use of NIEM should maintain compatibility with those standards, by doing things as the external standards intend. For example, Security Assertion Markup Language (SAML) uses attribute assertions, which each have a name and a value. In such a case, NIEM-conformant content should be carried as a set of values, with each name indicating a NIEM-conformant component. Specific rules and guidelines will be developed for each standard.

Overarching guidelines for using NIEM content in external standards will be developed. Such rules will be overridden by rules for specific standards. General rules will involve (1) focusing on elements as the prime semantic construct in NIEM-conformant data models, and (2) focusing on types as the prime structural entity. These guidelines will help maintain consistency in uses of NIEM content between various architectures.

Background, and Terminology

We use the term “schema component” for any object constructed by XML Schema. Schema components are specified by the XML Schema specification. They include attribute declarations, type definitions, etc. Some of these components may not be referenced from imported XML Schemas, and so are not concerns of this discussion. They include attribute uses (which are distinct from attribute declarations) and use of model groups (distinct from model group definitions).

From XML Schema Part 1: Structures 2d Ed
W3C Recommendation 28 October 2004

> [Definition:] **Schema component** is the generic term for the building blocks that comprise the abstract data model of the schema. [Definition:] An **XML Schema** is a set of schema components. There are 13 kinds of component in all, falling into three groups. The primary components, which may (type definitions) or must (element and attribute declarations) have names are as follows:
>
> - Simple type definitions
> - Complex type definitions
> - Attribute declarations
> - Element declarations
>
> The secondary components, which must have names, are as follows:
>
> - Attribute group definitions
> - Identity-constraint definitions
> - Model group definitions
> - Notation declarations
>
> Finally, the "helper" components provide small parts of other components; they are not independent of their context:
>
> - Annotations
> - Model groups
> - Particles
> - Wildcards
> - Attribute Uses

This document is concerned only with the use of components that may be referenced from imported namespaces. Such components may be defined in one schema and used in another, when the referencing schema imports the schema that defines the component. This specification also does not pay attention to Notations and Identity constraints. Specifically, this document supports the referencing of the following types of components from external namespaces:

- Simple type definitions
- Complex type definitions
- Attribute declarations
- Element declarations
- Attribute group definitions
- Model group definitions

NIEM Components

We use the term "NIEM Component" for a schema component from a namespace that is NIEM-conformant, which follows the rules defined by the NIEM Naming and Design Rules (NDR) for NIEM conformance. The NIEM NDR provides a profile of W3C XML Schema, along with additional constructs to support creating a data model. In order to be NIEM-conformant, a namespace must claim conformance, and must follow specific rules about structure, XML Schema feature usage, naming, and documentation.

NIEM conformance is determined at the namespace level, based on a reference schema for a particular namespace. To determine if a namespace is NIEM-conformant, the reference schema for the namespace is tested against a set of NIEM conformance rules. These rules include such things as:

(1) The schema must claim to be NIEM conformant.
(2) The schema must have a target namespace, over which the schema author has dominion.
(3) Schema components must be documented.
(4) Component documentation must take specific forms, including being supported with XML annotations from a NIEM-specific namespace, to support data modeling concepts.

External Components

We use the term "External Component" for a schema component from a namespace that does not follow the rules for NIEM conformance.

Examples of external, non-NIEM standards include:

- GML: Geography Markup Language. GML is a prime candidate for content that may be included in NIEM structures.
- XHTML: Extensible HyperText Markup Language. This language would likely be used for exchanging simple structured text.
- SAML: Security Assertion Markup Language. This is a likely language into which NIEM content will be embedded. Some SAML assertions will likely need to contain content defined by NIEM.

Goals and Requirements

There are numerous goals in defining and using adapter components:

(1) Ensure that content may be carried as defined by external standards, without modification. It would be bad to modify external content standards to make them "fit" NIEM. Instead, we should carry external content exactly as would be expected by external tools.
(2) Allow modeling efforts to define the granularity of use of external content as needed. Different groups require different levels of resolution into external standards. For example, one group may exchange a large block representing the geography of a large urban area, while another group may only exchange a single geographic point.

Both may be using the same external standard, but will need to use different parts, with different sizes, and different semantics.

(3) Ensure specific modeling efforts (e.g. NIEM core, domains, or IEPDs) are independent of a centralized process for use of external components. Developers should be able to immediately utilize standards required for their area of expertise.

(4) Provide points for harmonization. NIEM-conformant components that use external standards may be integrated into the core of NIEM. By using narrowly defined components with specific semantics, common uses that meet broad requirements may be identified and pulled up into the core of NIEM.

(5) Reduce coupling between NIEM schemas and external schemas. This method ensures that any given implementation uses external schemas through specific known points of access, without excessive dependence on deeply nested external components

Use of External Components in NIEM

Use of external schemas within NIEM should not require that each must be harmonized with components in or placed directly into NIEM. XML Schema provides namespaces to maintain semantic independence or uniqueness.

There are potentially two ways to use components from an external standard within the NIEM framework:

Methods of integrating external components

Profile the standard and insert its components into a NIEM namespace (potentially within Common).

Advantages:

(1) The "standard" components will become part of NIEM.
(2) The "standard" components will be registered with other NIEM components.
(3) The "standard" components will have precise semantics (because they ARE NIEM components).

Disadvantages:

(1) Anytime the standards change, then NIEM must potentially be changed.
(2) The "standard" components must be factored to conform to the NIEM NDR.
(3) The "standard" components may need to be harmonized with the NIEM components they potentially duplicate in whole or part (from a semantic perspective).
(4) The "standard" components may not be used in the same structural representation as the standard intended.
(5) Interoperability may require translations back to standard structures. (Tools expecting standard geospatial component structures may not recognize or work with those components if they have been re-structured per the NIEM NDR.)

(6) EACH standard that NIEM incorporates in this manner will result in the preceding disadvantages. Thus, insertion will require a lot of unnecessary extra work.

Provide a mechanism to reference and import the standard schemas/namespaces and use the components they contain within IEPDs constructed from NIEM components.

Advantages:

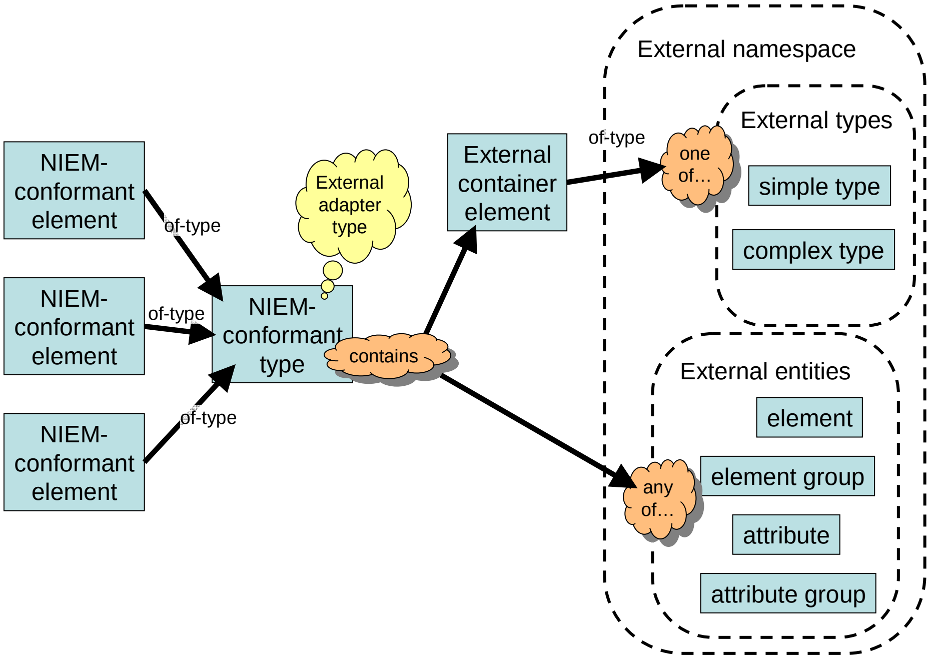
(1) Standards components are used within IEPDs in the same structure they were intended without the need to translate -- a potential boost to interoperability.
(2) Tools designed to recognize (parse) standard components from other namespaces will recognize these components.
(3) No other refactoring, integration, harmonization, or maintenance of these standards is required (even if the standards change).
(4) Components in preferred standards can still be registered with and discovered in a repository of NIEM components (requires storage of metadata about the standards).
(5) Subset schemas of these standards can be maintained and stored locally as required for use within IEPDs, a registry, etc.

Disadvantages:

(1) A NIEM structure (an adapter or container construct) will be required to encapsulate non-conforming components or schemas within an IEPD schema and to identify and preserve the semantics of those components.

Techniques

External components are encapsulated in NIEM-conformant components. This introduces the concept of “external adapter” types. An external adapter type is a NIEM-conformant XML Schema complex type that wraps a set of external content.



These adapter types and container elements are XML Schema components, and so are defined within the namespace of the schema currently being defined.

This document specifies two constructs, which contain external content. The first is the *external adapter type*. This type is a NIEM-conformant type that contains attributes and elements from external namespaces. The second is the *external container element*. The container element is used when an external namespace provides top-level types for use, but does not provide appropriate top-level elements. In such a case, create a container element of the externally-provided type. Container elements are defined in NIEM-conformant namespaces, are named differently than regular NIEM-conformant elements, and are used in a more restricted way.

Consistent with the fundamentals of NIEM, XML elements are used for semantics, and XML Schema types are used to contain necessary structures. Specific rules for definition of adapter components will take this approach, focusing on encapsulating external structures as NIEM-conformant types, within strongly-defined elements with specific semantics.

If an external type needs to be extended for use, such extension should be done **outside** a NIEM-conformant namespace. These structures are intended to encapsulate external content. They are not indented to introduce extensions and modifications to external content into NIEM-conformant namespaces. If an application schema needs to be constructed to conform to an external standard, the schema should be created in a userdefined namespace, outside the NIEM-conformant namespaces. Then, those external components should be referenced by NIEM-conformant external adapter types and external container elements, as specified below.

Details

This section contains rules for using external standards in NIEM. The section uses terminology specified by [XML Information Set (Second Edition), W3C Recommendation 4 February 2004, http://www.w3.org/TR/2004/REC-xml-infoset-20040204]. It also follows [XML Schema Part 1: Structures Second Edition, W3C Recommendation 28 October 2004, "http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/"].

The namespace prefix "i" is used in this specification as if bound to the namespace URI "http://niem.gov/niem/appinfo/0.3". This namespace is used by NIEM to describe information that occurs in the schema. Such information may be used by tools, to test conformance, and to support the data model definition of schema content.

Namespace Conformance

A namespace can be labeled as NIEM-conformant. Any namespace that is not NIEM-conformant is referred to as an external namespace. A namespace is NIEM-conformant if its reference schema follows NIEM conformance rules. A schema component must be in a NIEM-conformant namespace to be considered NIEM conformant. For any component of a schema to be conformant, the entire schema must be conformant. A NIEM-conformant schema must claim to be conformant. This occurs when the document element, the schema element, has a child annotation with a child appinfo with a child element `i:conformant` with the character child "true". In other words, the XPath "`/xsd:schema/xsd:annotation/xsd:appinfo/i:conformant`" has the value "`true`".

```
<xsd:schema ...>
  <xsd:annotation>
    <xsd:appinfo>
      <i:conformant>true</i:conformant>
    </xsd:appinfo>
  </xsd:annotation>
</xsd:schema>
```

This document only specifically addresses conformance issues for NIEM namespaces with respect to use of components from external namespaces.

Non-Schema Namespaces

An external namespace may be defined by a non-schema mechanism, such as DTD. In such a case, a *placeholder schema* would be created to represent the exact constructs referred to from the NIEM-conformant schema. A placeholder schema would not represent the deeper XML content of such namespaces. Instead, it would define placeholder elements and additional required constructs that are further defined by the non-XML Schema standard.

For example, XHTML 1.0, which has no normative XML Schema definition, may be considered an external namespace. XHTML defines a namespace, and numerous elements within that namespace. Were a NIEM-conformant schema specification to use the element "xhtml:ul" (an unordered list), it would use a reference. In order for schema validation to proceed normally, a schema would have to define that element. However, there is no such schema for Non-XML Schema specifications. The schema that is created to fulfill that role is the placeholder schema. Placeholder schemas should only represent the necessary components directly referred to from NIEM-conformant schemas.

Importing of External Namespaces

When NIEM namespaces are imported, the import statements are documented with a description of how the namespace is relevant to the namespace being defined. External (non-NIEM) namespaces should be documented with additional information, including:

1. An indication that the imported namespace is not NIEM-conformant.
2. The URI for a source of the reference schema for the namespace
3. Version information
4. Information about the body responsible for the standard, including:
    1. Contact information
    2. URI

Additional metadata will be defined, as the NIEM NDR is further defined. For the time being, the metadata should be included as documentation elements.

External Adapter Types

A NIEM external adapter type is a complex type that has the following qualities:

1. It is a special form of NIEM-conformant type. It may be used as the type of any NIEM-conformant element.
2. An adapter type should compose a single semantic entity. That is, the subparts of the type should appear together because they form the definition for some concept, not simply as a way of wrapping a block of external content.
3. An adapter type should be documented, as should any NIEM-conformant type.
4. It contains content from an external namespace, including:
    a. Attributes from an external namespace
    b. Attribute groups from an external namespace
    c. A single XSD sequence containing zero or more of:
        i. Elements from an external namespace
        ii. Model Groups from an external namespace. These are named groups of elements defined schemas.
        iii. *External container elements*, from a NIEM-conformant namespace. These are used when an external **type** must be used. They are defined below.
5. It must extend the "ComplexObjectType" from the NIEM structures namespace

6. It may not directly reference any other complex or simple types. Such types should be accessed via an external container element.
7. It may not directly reference other NIEM content. Apart from the "ComplexObjectType", all content of an external adapter type should be external.
8. The content it references may be from more than one external namespace.
9. Each referenced external component must be individually documented, describing the meaning of the external component

Additional annotations may be introduced as the NDR is developed.

An example of the simple case shows an adapter type directly referring to an external element:

```
<complexType name="PointType">
  <annotation>
    <documentation>
      SUMMARY OF TYPE GOES HERE
    </documentation>
  </annotation>
  <complexContent>
    <extension base="s:ComplexObjectType">
      <sequence>
        <element ref="gml:Point">
          <annotation>
            <documentation>
              DESCRIPTION OF EXTERNAL ELEMENT GOES HERE
            </documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

An alternate case occurs when **types** from an external standard need to be used, instead of elements.

External Container Elements

This specification introduces the term "External" as a suffix to element names in NIEM-conformant namespaces. An element with a name that ends in "External" is referred to as an *external container element*. Such an element is defined when a NIEM standard needs to reference XML Schema types from an external namespace.

If an external namespace defines elements that are appropriate for use, the elements should be referenced by external adapter types, and external container elements are unnecessary. External container elements are needed to create container elements for types from external namespaces.

An external container element has the following characteristics:

1. Its name ends in "External".
2. It is not a NIEM-conformant element.
3. It may only be referred to by *external adapter types*. It is an error for any other component to refer to an external container element.
4. The type of the element is a simple or complex type from an external namespace. The element definition may not reference any other external components.
5. An external container element may not specify a substitution group.

External container elements may not be referenced by standard conformant components. They may only be referenced by external adapter types.

Here is an example definition of an external container element:

```
<element name="PointExternal" type="gml:PointType">
  <annotation>
    <documentation>
      DESCRIPTION OF EXTERNAL TYPE GOES HERE
    </documentation>
  </annotation>
</element>
```

Note that the definition is very simple: it provides a container for an external type, and is clearly labeled as non-NIEM content by the suffix "External".

The external container element may be used by an adapter type, as the following example shows:

```
<complexType name="PointType">
  <annotation>
    <documentation>
      SUMMARY OF TYPE GOES HERE
    </documentation>
  </annotation>
  <complexContent>
    <extension base="s:ComplexObjectType">
      <sequence>
        <element ref="this:PointExternal"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

External container elements are not NIEM-conformant data model components. Instead, they create container for external types. They are clearly identified external by their names (suffixed with "External"). External elements (that come from non-NIEM namespaces) are clearly identified as external by their namespaces.

Registration of Components from External XML Standards with NIEM

When the use of data components from non-conforming standards is preferred within a NIEM IEPD, it is important to ensure that users are aware of such preferences. To highlight these preferences metadata associated with the external adapter types and external container elements can be registered and searched along with the NIEM components. NIEM tools will then discover and identify the preferred standards along with NIEM components. Search results will display the appropriate adapter type and container element components based on search criteria matching their metadata. Links to the content and documentation of each can also be presented. However, the NIEM architecture and tools will only know and use the metadata associated with an external standard as represented in the NIEM conforming components. NIEM will not attempt to understand or parse the content of the non-conforming standards that are contained within the conforming components that wrap them.

The current metadata registered with each NIEM component is used to search and navigate the NIEM. In addition to basic structural information such as type and parent type, there are also fields for both short summary definitions and longer full definitions. The capability to discover and use external standard components will depend greatly on the richness of the definitions provided for their external adapter types and external container types. Plans for NIEM will eventually incorporate and employ other forms of business context metadata for navigation and discovery. This metadata will also apply to external standards components.

As NIEM evolves, if it is discovered that the use of a particular preferred standard has become the norm, and that similarly defined NIEM components are being passed in favor of the standard, then such NIEM components can be deprecated and removed over time.

Summary

NIEM should treat all external standards with a consistent set of rules for external standards in an IEPD. Furthermore, NIEM should define how to use NIEM components in other schema standards. The NIEM architecture and tools can still facilitate the discovery of components required by IEPDs whether from external standards or from within NIEM without forcing external standards to conform to NIEM Naming and Design Rules.

Appendix: Examples

The files associated with this document provide a simple, validating example of the use of external standards with NIEM. The files are not intended to be used as-is, but show the pattern of using external XML Schemas. They use the geospatial schemas as provided by the DHS geospatial group

Files

The following files are included:

File	Description
geospatial-adapter.xsd	A schema that adapts components from GML and XLS. This schema builds types and elements in accordance with the Use of External Standards within NIEM paper.
instance1.xml instance2.xml	Sample instances that validate against the adapter schema
appinfo-0.3.xsd structures-0.3.xsd	Appinfo and structures schema, updated to include needed components
gml4niem.xsd xls4niem.xsd xlinks.xsd	External schemas, as provided by the DHS Geospatial group
instance.xsd	Simple top-level container schema

The sample schema uses an element substitution group in place of the "RoleOf" construct used in the DHS geospatial group's draft schemas.